toward increasing system throughput. Therefore read ahead buffer size may be limited using, for example, one of the following limiting relationships in conjunction with the appropriate respective resource model equation (17), (18) or (19) described above:

For single storage device case equation (17):

$$0.2 \leq (1\text{-Reserved\_Factor}) - (\Sigma_{i=1}^{Nov} \boldsymbol{P_i}) / (\text{TR}) \qquad (17\text{B})$$

For multiple storage device case under substantially balanced conditions equation (18):

$$0.2 \leq (1\text{-Reserved\_Factor}) - (\text{Skew/NoD}) * (\Sigma_{i=1}^{Nov} \boldsymbol{P_i}) / (\text{TR}) \qquad (18\text{B})$$

For multiple storage device case under substantially unbalanced conditions equation (19):

$$0.2 \leq (1 - \text{Reserved\_Factor}) - (\text{MaxAggRate\_perDevice /TR}) \qquad (19\text{B})$$

For example, the appropriate limiting relationship (17B), (18B) or (19B) may be substituted for the matching terms within the respective Resource Model Equation (17), (18) or (19) to limit the denominator of the left hand side of the respective Resource Model Equation to a limiting value of at least 0.2 so as to limit read-ahead size. In this regard, the limiting value of 0.2 is exemplary only, and may be varied as desired or necessary to fit particular applications.

It will be understood with benefit of this disclosure by those of skill in the art that the particular Resource Model Equations previously described are exemplary only, and that other equations, algorithms or other relationships may be employed using various other combinations of parameters described herein, and/or combinations of other parameters not explicitly described herein but that represent one or more information management system resource characteristics and/or operational characteristics as described elsewhere herein. In this regard, Resource Model Equations may be selected and/or customized to fit given information management system configurations. For example, the parameter Skew is optional for implementations that track and use the parameters MaxAggRate_perDevice and MaxNoV_perDevice to gain a more realistic view of workload distribution. Similarly, although the parameter B_Save appears in each of Resource Model Equations (17), (18) and (19), this parameter is not needed and may be removed from these equations in those embodiments where no explicit buffer sharing techniques are employed.

It will also be understood that the disclosed methods and systems for I/O resource management may be employed to manage I/O resources based on modeled and/or monitored I/O resource information in a wide variety of information management system configurations including, but not limited to, any type of information management system that employs a processor or group of processors suitable for performing these tasks. Examples include a buffer/cache manager (e.g., storage management processing engine or module) of an information management system, such as a content delivery system. Likewise resource management functions may be accomplished by a system management engine or host processor module of such a system. A specific example of such a system is a network processing system that is operable to process information communicated via a network environment, and that may include a network processor operable to process network-communicated information and a memory management system operable to reference the information based upon a connection status associated with the content.

Resource Modeling for Information Management System Implementations

In one exemplary embodiment, the disclosed methods and systems may be implemented in an information management system (*e.g.,* content router, content delivery system, *etc.*) to perform deterministic resource management in a storage management processing engine or subsystem module coupled to the information management system, which in this exemplary embodiment may act as an "I/O admission controller". Besides I/O admission control determinations, the disclosed methods and systems may also be employed in this embodiment to provide an estimation of read-ahead segment size. When implemented in conjunction with an integrated buffer/cache memory configuration such as described elsewhere herein, it is possible that introduction of a new viewer as described below may force a existing viewer to give up its interval in the cache and to come back to the I/O task pool. Thus, in such a case, admittance of a new viewer may result in admittance of two viewers into the I/O task pool.

Although this exemplary embodiment may be implemented in an information management system under any of the conditions described herein (*e.g.,* single storage device, multiple storage device/substantially balanced conditions, multiple storage device/substantially unbalanced conditions), the following discussion describes one example implementation in which an analytical-based resource model approach may be employed to manage I/O resources in an information management system where workload is substantially balanced or evenly distributed across multiple storage devices or groups of storage devices (*e.g.,* across disk drives or disk drive groups). It will be understood with benefit of this disclosure that a measurement-based resource model approach may be implemented in a similar manner, *e.g.,* under information management system I/O conditions where workload is not substantially balanced or evenly distributed across multiple storage devices or groups of storage devices, or under any other information management system I/O conditions (*e.g.,* substantially balanced workload across multiple storage devices) where implementation of such a measurement-based resource model is desired.

In this embodiment, an analytical-based resource model approach may be implemented by a storage processor such as storage management processing engine 105 of FIG. 1. In such a case, storage management processing engine 105 may be employed to monitor system I/O performance characteristics, including the total number of viewers supported by the system (NoV), the total number of viewers that are currently reading from storage devices 110 (NoV_IO), and the aggregated playback rates (*e.g.,* Sum of P_i values) during system operation.

As illustrated in FIG. 3A, a storage system 100 may start in step 300 with an existing cycle time T and an existing read-ahead size N_i, which remain unchanged in step 310 as long as no new viewer is introduced or no existing viewer is returned from cached state to I/O state. However, if in step 310 a new viewer is introduced or an existing viewer is returned from cached